Theory of a unified field: standard file formats and subroutine libraries for diffraction imaging

Jan Steinbrener[1], Andrew Stewart[1], Xiaojing Huang[1], Huijie Miao[1], David Shapiro[2], Chris Jacobsen[1]

[1]Dept. Physics & Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, USA
[2]Advanced Light Source, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Within a generation, protein crystallography has gone from tens of structures solved per year to thousands. In the same period, single particle methods in cryo electron microscopy have moved from being the subject of development in a few labs to something that is widely applied by many researchers. Advances in both of these fields have been due to a number of factors, but one of them is the adoption of a suite of highly developed programs and subroutine libraries which are able to work with common file formats.

We have begun to develop some interoperable capabilities between diffraction microscopy data handling and reconstruction software developed at Stony Brook, Lawrence Livermore National Laboratory, Lawrence Berkeley National Laboratory, and Cornell University. With this software (made available through a CVS or concurrent versioning system archive), we are able to automate the collection of data, combine raw diffraction patterns, run iterative reconstructions in 2D and 3D, and display the results. We propose here to increase the degree of interoperability, and reduce the work of building diffraction reconstruction software, by encouraging the community to contribute to the development of certain common standards for software. By using the widely-available, platform-independent HDF5 hierarchical data file format with a set of specific groups for different data functions, it will be possible to pass data files among different reconstruction programs to compare their success in reconstruction. By using a subroutine library written to carry out basic array operations such as FFTs, addition/subtraction, and array masking, it should be possible to develop C language programs that will compile and run both on single-processor laptops as well as MPI-enabled computational clusters. We describe our present efforts to develop file and array operation subroutines, and encourage community participation in developments towards common software structures.